Babraham Bioinformatics

Exercises:
Processing ChIP-Seq data

Version 2018-03-3

Licence

Introduction

In this session we will go through how to process a small ChIP-Seq experiment. This will include:

- Quality control of raw sequence data
- Mapping to a reference genome with bowtie2
- Collation of raw and mapped QC with MultiQC
- Visual inspection and exploration of the mapped data with SeqMonk

Software

The software which will be used in this session is listed below. Software which requires a linux environment is indicated by an asterisk*:

- Bowtie2* (http://bowtie-bio.sourceforge.net/bowtie2/)
- Samtools (http://www.htslib.org/)
- FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
- MultiQC (http://multiqc.info/)
- SeqMonk (http://www.bioinformatics.babraham.ac.uk/projects/seqmonk/)

Data

The data in this practical comes from ENCODE accession ENCSR260AKX
C. elegans genome data and GTF models come from Ensembl
(https://www.ensembl.org/Caenorhabditis_elegans/)

Exercise 1: Quality Control

To start with we are going to use the FastQC program to look at the basic properties of the 3 fastq files we are going to process.

Move into the folder which contains the sequence data

```
cd ChIP-Data/Worm_ChIP_Mapping
```

Run fastqc on all of the files.

```
fastqc *fastq.gz
```

If you run `ls` you should be able to see several new HTML files generated by fastqc. You can open these in firefox to look at them.

```
firefox *html &
```

(the `&` on the end puts the firefox program into the background, so it will keep running but allow you to enter more commands into the shell).

You can have a look through the QC results to try to answer the following questions:

- Did any of the QC modules trigger a warning or alert condition?
- Do the base call qualities provided by the sequencer suggest the data is high quality, or might it benefit from being quality trimmed?
- Are there any consistent sequence biases in the data?
- Is there any suggestion of the presence of adapter sequence which might need to be removed?
- Does the duplication level look as you'd expect for ChIP-Seq data?

Exercise 2: Mapping and QC

For some data sets it would make sense to do adapter and quality trimming. Why is that not necessary in this case?

Step 2.1 Creating a Genome Index

You will have been provided with a worm genome sequence in a file called `Caenorhabditis_elegans.WBcel235.fa`. Before you can run bowtie2 you will need to create an index file from this sequence which bowtie2 can then use.

```
bowtie2-build Caenorhabditis_elegans.WBcel235.fa celegans
```

The first option you supply here is just the name of the fasta file which contains your genome sequence. The second option is the base name for the index. This is the name you will need to supply when you run hisat2, and you should also now be able to see the following files in your directory:

```
celegans.1.bt2          celegans.2.bt2          celegans.3.bt2
celegans.4.bt2          celegans.rev.1.bt2      celegans.rev.1.bt2
```

Step 2.2 Running bowtie2

Since we're going to be mapping 4 files and the command we're going to construct is reasonably complex, we've provided you with a small script which will automate this called `map_reads.sh`. You are going to run this script (which we'll get to in a little bit), but we also want to show you what's in it so you can see what it's doing. The contents of this script are shown below. You do **NOT** need to type this in to your console, it's just to show you what the script will be doing.

```
for i in *fastq.gz
do
# Print some information
echo "Mapping $i"
echo "Running command: bowtie2 -p 2 -x celegans -U $i 2>$i.log | samtools
view -bS - > $i.bam"

# Run the mapping command
bowtie2 -p 2 -x celegans -U $i 2>$i.log | samtools view -bS - > $i.bam

# Show the contents of the log file which was generated.
cat $i.log
done
```

It's simply a small loop which runs a command similar to the one shown below for each file

```
bowtie2 -x celegans -U data.fq.gz 2> data.fq.gz.log | samtools view -bS - >
data.fq.gz.bam
```

The options in this command are as follows:

`-x` : Specifies the index file to use – this is the basename for the index which you created in step 2.1

`-p 2` : This tells hisat2 that it can use 2 CPU cores on the machine. Short read mapping scales very efficiently across cores so if you have more cores available then you should use them to speed up your analysis.

`-U` : Specifies the fastq file of sequence reads to map. If you had paired end data and therefore 2 input files you would use `-1` and `-2` to specify the two files instead of `-U`

`| samtools view -bS - >` : The final part of the command actually sends (pipes) the output of bowtie2 into another program called samtools. The purpose of this is to compress the raw sam format output from bowtie2 into the more compact bam format.

To run the mapping run:

```
./map_reads.sh
```

The mapping will take a few minutes to complete. Once it's finished you should see the mapping statistics printed in the console, and you should see the bam file of aligned reads. You can see the set of files which have been generated by running:

```
ls -h
```

Have a look through and make sure you're happy that you know what each of the files is and which program produced them.

Step 2.3 Running multiqc

To get a better overall picture of what's happened to all of our samples we are going to use multiqc to collate the QC and mapping statistics into a simple single report which we can view to assess any problems with the data.
To run multiqc you simply run:

```
multiqc .
```

(note the dot at the end – this just says to process the files found in the current directory)

This should create a file called `multiqc_report.html` which you can view in firefox as before.

```
firefox multiqc_report.html &
```

Have a look through the combined statistics for the 3 samples and see if there are any of the metrics where they differ. If you find any differences, can you think of why they might legitimately be different for good biological reasons rather than bad technical ones?